Lecture 10: Risk Management

General ideas about Risk

Risk Management
- Identifying Risks
- Assessing Risks

Case Study:
- Mars Polar Lander

Risk Management

About Risk
- Risk is "the possibility of suffering loss"
- Risk itself is not bad, it is essential to progress
- The challenge is to manage the amount of risk

Two Parts:
- Risk Assessment
- Risk Control

Useful concepts:
- For each risk: Risk Exposure
  \[ RE = p(\text{unsat. outcome}) \times \text{loss(unsat. outcome)} \]
- For each mitigation action: Risk Reduction Leverage
  \[ RRL = \frac{RE_{\text{before}} - RE_{\text{after}}}{\text{cost of intervention}} \]

Principles of Risk Management

Global Perspective
- View software in context of a larger system
- For any opportunity, identify both:
  - Potential value
  - Potential impact of adverse results

Forward Looking View
- Anticipate possible outcomes
- Identify uncertainty
- Manage resources accordingly

Open Communications
- Freely flow information at all project levels
- Value the individual voice
- Unique knowledge and insights

Integrated Management
- Project management is risk management

Continuous Process
- Continually identify and manage risks
- Maintain constant vigilance

Shared Product Vision
- Everybody understands the mission
- Common purpose
- Collective responsibility
- Shared ownership
- Focus on results

Teamwork
- Work cooperatively to achieve the common goal
- Pool talent, skills and knowledge

Continuous Risk Management

Identify:
- Search for and locate risks before they become problems
- Systematic techniques to discover risks

Analyze:
- Transform risk data into decision-making information
- For each risk, evaluate:
  - Impact
  - Probability
  - Timeframe
- Classify and Prioritize Risks

Plan
- Choose risk mitigation actions

Control
- Correct for deviations from the risk mitigation plans

Communicate
- Share information on current and emerging risks

Source: Adapted from SEI Continuous Risk Management Handbook
Fault Tree Analysis

Event that results from a combination of causes

- Wrong or inadequate treatment administered
- Vital signs erroneously reported as exceeding limits
- Vital signs exceed critical limits but not corrected in time
- Frequency of measurement too low
- Computer fails to raise alarm
- Nurse does not respond to alarm
- Computer does not read within required time limits
- Human sets frequency too low
- Sensor failure
- Nurse fails to input them or does so incorrectly
- etc.

Risk Assessment

- Quantitative:
  - Measure risk exposure using standard cost & probability measures
  - Note: probabilities are rarely independent

- Qualitative:
  - Develop a risk classification matrix:

<table>
<thead>
<tr>
<th>Likelihood of Occurrence</th>
<th>Very likely</th>
<th>Possible</th>
<th>Unlikely</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5) Loss of Life</td>
<td>Catastrophic</td>
<td>Catastrophic</td>
<td>Severe</td>
</tr>
<tr>
<td>(4) Loss of Spacecraft</td>
<td>Catastrophic</td>
<td>Severe</td>
<td>High</td>
</tr>
<tr>
<td>(3) Loss of Mission</td>
<td>Severe</td>
<td>Severe</td>
<td>High</td>
</tr>
<tr>
<td>(2) Degraded Mission</td>
<td>High</td>
<td>Moderate</td>
<td>Low</td>
</tr>
<tr>
<td>(1) Inconvenience</td>
<td>Moderate</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Top 10 Development Risks (± Countermeasures)

- Personnel Shortfalls
  - Use top talent
  - Team building
  - Training
- Unrealistic schedules/budgets
  - Multisource estimation
  - Designing to cost
  - Requirements scrubbing
- Developing the wrong Software functions
  - Better requirements analysis
  - Organizational/operational analysis
- Developing the wrong User Interface
  - Prototypes, scenarios, task analysis
- Gold Plating
  - Requirements scrubbing
  - Cost benefit analysis
  - Designing to cost
- Continuing stream of reqts changes
  - High change threshold
  - Information hiding
  - Incremental development
- Shortfalls in externally furnished components
  - Early benchmarking
  - Inspections, compatibility analysis
- Shortfalls in externally performed tasks
  - Pre-award audits
  - Competitive designs
- Real-time performance shortfalls
  - Targeted analysis
  - Simulations, benchmarks, models
- Straining computer science capabilities
  - Technical analysis
  - Checking scientific literature

Case Study: Mars Polar Lander

- Launched
  - 3 Jan 1999
- Mission
  - Land near South Pole
  - Dig for water ice with a robotic arm
- Fate:
  - Arrived 3 Dec 1999
  - No signal received after initial phase of descent
- Cause:
  - Several candidate causes
  - Most likely is premature engine shutdown due to noise on leg sensors
What happened?

Investigation hampered by lack of data

- spacecraft not designed to send telemetry during descent
- This decision severely criticized by review boards

Possible causes:

- Lander failed to separate from cruise stage (plausible but unlikely)
- Landing site too steep (plausible)
- Heatshield failed (plausible)
- Loss of control due to dynamic effects (plausible)
- Loss of control due to center-of-mass shift (plausible)
- Premature Shutdown of Descent Engines (most likely!)
- Parachute drapes over lander (plausible but unlikely)
- Backshell hits lander (plausible but unlikely)

Premature Shutdown Scenario

Cause of error

- Magnetic sensor on each leg senses touchdown
- Legs unfold at 1500m above surface
- Transient signals on touchdown sensors during unfolding
- Software accepts touchdown signals if they persist for 2 timeframes
- Transient signals likely to be long enough on at least one leg

Factors

- System requirement to ignore the transient signals
- But the software requirements did not describe the effect
- s/w designers didn’t understand the effect, so didn’t implement the requirement
- Engineers present at code inspection didn’t understand the effect
- Not caught in testing because:
  - Unit testing didn’t include the transients
  - Sensors improperly wired during integration tests (no touchdown detected)
- Full test not repeated after re-wiring

Result of error

- Engines shut down before spacecraft has landed
- When engine shutdown s/w enabled, flags indicated touchdown already occurred
- Estimated at 40m above surface, travelling at 13 m/s
- Estimated impact velocity 22m/s (spacecraft would not survive this)
- Nominal touchdown velocity 2.4m/s

Learning the Right Lessons

- Understand the Causality
  - Never a single cause; usually many complex interactions
  - Seek the set of conditions that are both necessary and sufficient...
    - to cause the failure

- Causal reasoning about failure is very subjective
  - Data collection methods may introduce bias
    - e.g. failure to ask the right questions (or provide appropriate response modes)
  - Human tendency to over-simplify
    - e.g. blame the human operator
    - e.g. blame only the technical factors

“In most of the major accidents of the past 25 years, technical information on how to prevent the accident was known, and often even implemented. But in each case... [this was] negated by organisational or managerial flaws.” (Leveson, Safeware)
Is there an existing “Safety Culture”?

→ Are overconfidence and complacency common?
  - The Titanic effect - “it can’t happen to us!”
  - Do managers assume it’s safe unless someone can prove otherwise?

→ Are warning signs routinely ignored?
  - What happens to diagnostic data during operations?
  - Does the organisation regularly collect data on anomalies?
  - Are all anomalies routinely investigated?

→ Is there an assumption that risk decreases?
  - E.g. Are successful missions used as an argument to cut safety margins?

→ Are the risk factors calculated correctly?
  - E.g. What assumptions are made about independence between risk factors?

→ Is there a culture of silence?
  - What is the experience of whistleblowers? (Can you even find any?)

Failure to manage risk

Science (functionality)  
Risk Only variable

Schedule  
Fixed  
Cost  
Fixed

Launch Vehicle  
Fixed  
(Some Relief)

Inadequate Margins

Science (functionality)

See http://www.nasa.gov/newsinfo/marsreports.html

Summary

→ Risk Management is a systematic activity
  - Requires both technical and management attention
  - Requires system-level view
  - Should continue throughout a project

→ Techniques exist to identify and assess risks
  - E.g. fault tree analysis
  - E.g. Risk assessment matrix

→ Risk and Requirements Engineering
  - Risk analysis can uncover new requirements
    - Especially for safety-critical or security-critical applications
  - Risk analysis can uncover feasibility concerns
  - Risk analysis will assist in appropriate management action